UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS TRINDADE OU CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Otto Menegasso Pires

A quantum heuristic for the School Timetabling Problem

Florianópolis
2021

Otto Menegasso Pires

A quantum heuristic for the School Timetabling Problem

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Ciência da Computação.
Supervisor: Prof. Rafael de Santiago, Dr.
Coorientadora: Profa. Jerusa Marchi, Dra.

Florianópolis
2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pires, Otto Menegasso
A quantum heuristic for the School Timetabling Problem / Otto Menegasso Pires ; orientador, Rafael de Santiago, coorientadora, Jerusa Marchi, 2021.
47 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2021.

Inclui referências.

1. Ciência da Computação. 2. qaoa. 3. quantum computing. 4. school timetabling. 5. timetabling problem. I. de Santiago, Rafael. II. Marchi, Jerusa. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

Otto Menegasso Pires

A quantum heuristic for the School Timetabling Problem

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Franklin de Lima Marquezino, Dr.
Universidade Federal do Rio de Janeiro - UFRJ

Profa. Gisele Bosso de Freitas, Dra.
Universidade Estadual da Região Tocantina do Maranhão - UEMASUL

Prof. Rafael Augusto de Melo, Dr.
Universidade Federal da Bahia - UFBA

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Ciência da Computação.

Documento assinado digitalmente
PATRICIA DELLA MEA PLENTZ
Data: 10/12/2021 03:03:45-0300
CPF: 959.243.700-97
Verifique as assinaturas em https://v.ufsc.br

Coordenação do Programa de
Pós-Graduação

Documento assinado digitalmente
Rafael de Santiago
Data: 09/12/2021 19:39:45-0300
CPF: 041.277.019-94
Verifique as assinaturas em https://v.ufsc.br

Prof. Rafael de Santiago, Dr.
Supervisor:

Documento assinado digitalmente
Jerusa Marchi
Data: 09/12/2021 20:27:29-0300
CPF: 026.208.419-84
Verifique as assinaturas em https://v.ufsc.br

Profa. Jerusa Marchi, Dra.
Coorientadora

Florianópolis, 2021.

AGRADECIMENTOS

Eu gostaria de agradecer ao Rafael de Santiago por ter confiado em mim e aceitado orientar o meu mestrado e à Jerusa Marchi por ter me coorientado na graduação e agora novamente no mestrado e por acompanhar desde o início a minha trajetória na computação quântica.

Eu quero agradecer imensamente ao Evandro Chagas por todo auxílio que ele me deu durante a elaboração desse trabalho, sem a sua ajuda a minha pesquisa não teria sido bem-sucedida.

E finalmente, quero agradecer aos membros do Grupo de Computação Quântica da UFSC e a todos os amigos e família que me acompanharam durante essa jornada.

“They both savoured the strange warm glow of being much more ignorant than ordinary people, who were only ignorant of ordinary things.”
(Terry Pratchett, Equal Rites)

RESUMO

School timetabling é uma variação do Problema de Alocação de Horários que busca uma alocação periódica de aulas para alunos e professores de uma escola, que deve seguir um conjunto de restrições fortes e fracas. *Timetabling* é um problema NP-Hard e, por causa de sua dificuldade, o uso de heurísticas para solucionar o problema é uma prática comum. Quando apenas as restrições fortes são consideradas, o problema *timetabling* pode ser reduzido para a coloração de grafos e a similaridade entre esses problemas tem motivado o uso de heurísticas para coloração de grafos como um meio de solucionar o problema de *timetabling*. Esse trabalho propõe uma otimização em dois passos para solucionar o problema de *school timetabling*, onde no primeiro passo o problema é reduzido para coloração de grafos e o circuito quântico do *Quantum Approximate Optimization Algorithm* (QAOA) para a solução do problema do número cromático é usado para resolver as restrições fortes e no segundo passo o processo de otimização clássico do QAOA é usado para solucionar as restrições fracas. A heurística foi testada em instâncias de *benchmark* do banco de dados do XHSTT. Os circuitos quânticos do trabalho possuíam até 189 qubits e foram simulados em um ambiente sem ruído. Essa pesquisa e seus resultados são um trabalho seminal no uso do QAOA como heurística para o problema de *timetabling*.

Palavras-chave: qaoa, computação quântica, timetabling problem, school timetabling

RESUMO EXPANDIDO

Introdução

Timetabling pode ser visto como um problema de otimização combinatória. Esses problemas são sobre como encontrar a solução ótima para uma atribuição de valores a variáveis discretas de acordo com alguns critérios, chamados de restrições fortes e fracas. As restrições fortes definem a viabilidade da solução e as restrições fracas definem sua qualidade. Para o *School Timetabling Problem*, a solução ótima é definida como um horário viável que satisfaça a quantidade máxima de restrições fracas.

Considerando-se apenas as restrições fortes, o problema de *Timetabling* pode ser representado como um problema de coloração dos vértices de um grafo. Colorir um grafo é equivalente a encontrar a quantidade de períodos de tempo necessários para agendar os eventos sem conflito de tempo. A similaridade entre *Timetabling* e coloração de grafos tem motivado o uso de heurísticas de coloração de grafos para resolver o problema de *Timetabling*.

Os computadores quânticos atuais estão sob o regime *Noise Intermediate-Scale Quantum* (NISQ). Eles têm conectividade de circuito limitada e baixa qualidade de qubits. Além disso, seu tamanho varia de 50 a 100 qubits. Essas limitações impedem a implementação de alguns dos algoritmos quânticos mais notáveis. Porém, esse regime atual é suficiente para rodar algoritmos quânticos difíceis de simular em hardware clássico, alcançando a chamada supremacia quântica. No entanto, uma importante questão em aberto é se os computadores quânticos podem fornecer um aumento de velocidade, o que muitas vezes é chamado de vantagem quântica, na solução de problemas de otimização combinatória classicamente difíceis, já que muitos problemas do mundo real se enquadram nesta categoria. Devido à sua ampla aplicabilidade, o *Quantum Approximate Optimization Algorithm* (QAOA) já foi usado como uma heurística para resolver esta classe de problemas.

Assim, este trabalho levanta as seguintes questões: O QAOA pode ser usado como uma heurística para o School Timetabling Problem modelado como um Problema de Coloração de Grafos? Que tipo de vantagem essa heurística pode oferecer ao problema?

Objetivos

Objetivo Geral: Este trabalho tem como objetivo propor uma nova heurística quântica para o *School Timetabling Problem* e comparar seu desempenho com as heurísticas clássicas atualmente conhecidas.

Objetivos Específicos:

- Objetivo 1: Estudar o estado da arte do QAOA;

- Objetivo 2: Estudar o estado da arte das heurísticas para o *School Timetabling Problem*;
- Objetivo 3: Compilar a estrutura QAOA descrita em (HADFIELD, 2018) para um circuito quântico;
- Objetivo 4: Comparar a eficiência da solução proposta com o estado da arte das heurísticas clássicas.

Metodologia

A pesquisa aqui apresentada é quantitativa porque propõe uma comparação relacionada à qualidade dos métodos computacionais desenvolvidos durante o trabalho.

1. Estudo do estado da arte do QAOA;
2. Análise do framework QAOA para resolução de problemas com restrições fortes e fracas;
3. Compilação do framework QAOA para a linguagem Ket;
4. Estudo do estado da arte em heurísticas para *Timetabling*;
5. Comparação entre QAOA e outras heurísticas;
6. Redação de artigos científicos;
7. Redação da tese

Resultados

Foi desenvolvido uma nova heurística quântica para o *School Timetabling Problem*, com seus primeiros resultados publicados em (PIRES; SANTIAGO; MARCHI, 2021). Todo o código utilizado pelo trabalho pode ser encontrado em um repositório no GitLab(PIRES, 2021). As implementações deste trabalho também forneceram um *benchmark* para o Ket Bitwise Simulator.

Discussão e Considerações

Este trabalho propõe o uso do QAOA como heurística para a solução do *School Timetabling Problem*. Foi desenvolvido um processo de otimização de dois estágios, onde no primeiro estágio o circuito quântico QAOA para o problema da coloração mínima de grafos otimiza as restrições fortes do cronograma e no segundo estágio o laço de otimização clássico do QAOA otimiza as restrições fracas. Consideramos esta pesquisa e seus resultados um trabalho seminal no uso de QAOA como uma heurística para o problema de *Timetabling*.

O método foi testado usando três instâncias diferentes. A instância Denmark-Smallschool do *Dataset* XHSTT e duas instâncias derivadas, Den-Aux e Den-5. Os circuitos quânticos foram simulados em um ambiente sem ruído usando o *Parallel Bitwise Simulator* para a linguagem de programação quântica Ket (DA ROSA; DE SANTIAGO, 2021). As instâncias exigiram até 189 qubits para serem simuladas.

A heurística não conseguiu otimizar a instância original Denmark-Smallschool devido ao misturador do QAOA ser incapaz de alcançar soluções diferentes. É possível que este comportamento indique que a heurística também terá dificuldades em otimizar instâncias difíceis do problema de *Timetabling*. Em relação às instâncias derivadas, a heurística não foi capaz de encontrar resultados melhores do que o estado inicial para a instância Den-Aux, no entanto, para a instância Den-5, a heurística foi capaz de convergir para um resultado otimizado com o parâmetro de profundidade $p = 2$. Isso pode indicar o potencial de usar essa heurística em computadores quânticos atuais, pois a profundidade necessária para o QAOA é baixa.

Palavras-chave: qaoa, computação quântica, timetabling problem, school timetabling

ABSTRACT

School timetabling is a variation of the Timetabling problem that searches for a periodic scheduling of lessons for classes and teachers of a school, that must meet a set of hard and soft constraints. Timetabling is an NP-Hard problem and because of its difficulty, the use of heuristics to address it is a common practice. When only the hard constraints are considered, the timetabling problem can be reduced to graph vertex coloring and the similarity between both problems has motivated the use of graph coloring heuristics as a means to tackle the timetabling problem. We propose to tackle the school timetabling problem by applying a Two-stage optimization, where in the first stage we reduce it to a graph coloring problem and use the Quantum Approximate Optimization Algorithm (QAOA) quantum circuit for solving the chromatic number problem to address the hard constraints and on the second stage we address the soft constraints of the timetabling problem by using the classical optimization process of QAOA. We tested our heuristic using benchmark instances from the XHSTT dataset and we simulated quantum circuits up to 189 qubits in a noiseless environment. We consider this research and its findings a seminal work in using QAOA as a heuristic for the timetabling problem.

Keywords: qaoa, quantum computing, school timetabling, timetabling problem

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

1.1 Timetabling Problem

Timetables are crucial to many organizations. Public transportation, schools, and universities regularly use timetables to organize time. In fact it is difficult to imagine an organized and modern society that does not use them. The problem of constructing a timetable is considered NP-complete (EVEN; ITAI; SHAMIR, 1976) and despite its widespread use, it is unknown a polynomial-time exact method to construct timetables. Factors such as the limitation of resources, e.g. people or time, or the many aspects that differentiate the quality of a timetable increase its difficulty. Many works have addressed this problem (HILTON, 1981; DE WERRA, 1985; COLBOURN; OORSCHOT, 1989; SCHAERF, 1999; BELIGIANNIS et al., 2008; RAGHAVJEE; PILLAY, 2009; PILLAY, Nelishia, 2013; RIBIĆ; TURČINHOŽIĆ; MURATOVIĆ-RIBIĆ, 2015; VEENSTRA; VIS, 2016) and the need for better timetables solutions still motivates further research nowadays (TURCINHODZIC et al., 2019; SAVINIEC et al., 2020; TAN et al., 2021).

Timetabling can be seen as a combinatorial optimization problem (WERRA; KOBLER, 2014). These problems are about finding the optimal solution to an assignment of discrete values to variables according to some criteria, called hard and soft constraints. The hard constraints define the feasibility of the solution and the soft constraints define its quality. For the school timetabling problem, the optimal solution is defined as a feasible timetable that satisfies the maximum amount of soft constraints. Finding an exact solution to the timetabling problem is considered NP-Hard (COOPER; KINGSTON, 1996). Because of that, meta-heuristics have been widely employed to tackle timetabling problems, instead of using exact methods (FONSECA; SANTOS; CARRANO, 2016).

Considering only the hard constraints, the timetabling problem can be represented as a graph vertex coloring problem. A graph coloring is obtained by choosing a color for each node of the graph in a way that none of its neighbors share its color and serves as a general model for conflict resolution. When applied to the school timetabling problem, a conflict graph is created where each edge represents a time conflict of a student or a teacher. Coloring the conflict graph is equivalent to find the length of time periods required to schedule the lectures without time conflict. The similarity between timetabling and graph coloring has motivated the use of graph coloring heuristics to tackle the timetabling problem (DE WERRA, 1985; BUDIONO; WONG, 2012).

1.2 Quantum computation and quantum algorithms

Current state-of-the-art quantum computers are referred to as Noise Intermediate-Scale Quantum (NISQ) computers (PRESKILL, 2018). They have limited circuit con-

nectivity and qubit quality. Moreover, their size ranges from 50 to 100 qubits. These limitations prevent the implementation of some of the most notable quantum algorithms (GIDNEY; EKERÅ, 2021). However, this current regime is enough to run quantum algorithms which are hard to simulate on classical hardware, achieving the so called quantum supremacy (ARUTE et al., 2019).

Considering the scarcity of the available resources, the key question is how to make the best use of NISQ devices to run algorithms capable of solving real world problems. The most promising candidate for running on current quantum computers are the class of Variational Quantum Algorithms (VQA) (CEREZO et al., 2020). VQA are hybrid quantum-classical algorithms that optimize a cost function encoded by the expectation value of a parametrized quantum circuit, sometimes referred as an *ansatz*. This ansatz is trained in a hybrid quantum-classical loop, where a classical optimization algorithm make functions calls to a quantum computer to optimize the expectation value over the quantum circuit's parameters. In contrast to quantum algorithms developed for the fault-tolerant era, VQA are capable of keeping the quantum circuit depth shallow, hence mitigating noise, by leveraging the power of classical optimizers to train the parameters of its quantum circuit. Two examples of VQA are the Quantum Approximate Optimization Algorithm (QAOA) (FARHI; GOLDSTONE; GUTMANN, 2014) and the Variational Quantum Eigensolver (VQE) (PERUZZO et al., 2014).

Nevertheless, an important open question is whether quantum computers can provide an speedup, which is often called quantum advantage, in solving classically hard combinatorial optimization problems, as many real world problems fall in this category. Due to its broad applicability, QAOA was used before as an heuristic to address this class of problems, for example the Vehicle Routing Problem (UTKARSH; BEHERA; PANIGRAHI, 2020) and the Tail Assignment Problem (VIKSTÅL et al., 2020), in both cases the instances used at most 25 qubits. Although it is an emerging topic, for the best of our knowledge, it is lacking an investigation over classical optimization problems. So, we rise the following questions:

Can QAOA be used as an heuristic for the School Timetabling Problem modeled as a Graph Coloring Problem? Which kind of advantage this heuristic can offer to the problem?

1.3 Main Objective

Our work has the goal to propose a new quantum heuristic for the School Timetabling Optimization Problem and to compare its performance against current known classical heuristics.

Research question: Can QAOA be used as an heuristic for the School Timetabling Problem modeled as a Graph Coloring Problem? Which kind of advantage this heuristic can offer to the problem?

1.4 Objectives

- Objective 1: To study the state-of-the-art of QAOA;
- Objective 2: To study the state-of-the-art of Timetabling Optimization Problem heuristics;
- Objective 3: To compile the QAOA framework described in (HADFIELD, 2018) to a quantum circuit;
- Objective 4: To compare the efficiency of our solution against state-of-the-art classical heuristics.

1.5 Scope

We explore the use of Quantum Approximate Optimization Algorithm as a heuristic to address the School Timetabling Problem when modeled as a Graph Coloring Problem. We use benchmark instances for High School Timetabling from (POST et al., 2012). We choose the Parallel Bitwise Simulator for the Ket Quantum Programming Language (DA ROSA; DE SANTIAGO, 2021) to run our quantum circuit because it is able to simulate circuits with more qubits than other available simulators. We simulate the quantum circuit in a noiseless environment and required up to 189 qubits to simulate our quantum circuit.

1.6 Methodology

The research presented here is quantitative because it proposes a comparison related to the quality of the computational methods developed during the work.

1. Study of the state-of-the-art of QAOA;
2. Analysis of the QAOA framework for solving problems with hard and soft constraints;
3. Compilation of the QAOA framework to the Ket Language;
4. Study of the state-of-the-art in Timetabling heuristics;
5. Comparison between QAOA and other heuristics;
6. Scientific paper writing;
7. Thesis writing.

1.7 Contributions

We developed a new quantum heuristic for the school timetabling problem, with our first results published in (PIRES; SANTIAGO; MARCHI, 2021). All code used in our work can be found in a repository on GitLab (PIRES, 2021). Our implementations also provided a benchmark for the Ket Bitwise Simulator.

1.8 Overview

This work is structured as follows: chapter 2 introduces the school timetabling problem. Chapter 3 describes the QAOA algorithm and its variation, the Quantum Alternating Operator Ansatz. Chapter 4 describes our proposed heuristic. Chapter 5 shows our results. Finally, the concluding remarks are in Chapter 6.

2 SCHOOL TIMETABLING PROBLEM

The timetabling problem consists in scheduling a sequence of events in a prefixed period of time satisfying a set of constraints of various types. This problem varies according to the institution involved and the type of constraints required by the timetable. There are three main classes of problems (SCHAERF, 1999):

- School timetabling: periodic scheduling of lessons for classes and teachers of a school to a limited number of available time periods, avoiding time conflicts;
- Course timetabling: periodic scheduling of lessons from a set of university courses to a number of fixed time periods and rooms, minimizing the overlaps of lessons from courses having the same students;
- Examining timetabling: scheduling of exams of university courses, avoiding overlaps of exams having common students, locally spreading students as much as possible. The Examining timetabling differentiates itself from the Course timetabling because its scheduling is not periodic, and is restricted to a single day or two of the month instead of the whole week (AZIZ; AIZAM, 2018).

In this work we chose to focus on school timetabling.

2.1 Terminology

The terminology used to address the School Timetabling Problem is not consistent across different studies. In order to avoid misconceptions, we define the terms used in this work as follows:

- A *class* refers to a group of students that are taught a particular subject simultaneously.
- An *event group* is a set of events that share some characteristic. For example, all the lessons given by a specific teacher.
- A *lesson* refers to a particular subject being taught to a class by a teacher. A lesson is comprised of a class, teacher and a room which must be scheduled in a time period. In some cases a lesson may consist of only a class and a teacher. Also referred as an *event*.
- A *time group* is a set of time periods and it usually refers to a specific weekday.
- A *time period* is a timetable slot in which a lesson can be scheduled.
- A *resource* refers to any entity involved in a lesson. The standard resources are a class, teacher and the room in which the lesson is held.

2.2 Problem Definition

The School Timetabling Problem is defined in terms of the available teachers, students, number of lessons to be taught by teachers to specific students, and a set of constraints. Students are usually grouped into classes prior to the timetable construction process. A timetable is constructed by assigning lessons to time periods while respecting the predefined set of constraints (FONSECA; SANTOS; CARRANO, 2016; TASSOPOULOS; ILIOPOULOU; BELIGIANNIS, 2020).

Constraints can be one of two types, hard or soft. Hard constraints must be met in order for the timetable to be possible. A timetable meeting all the hard constraints of a problem is called a feasible timetable. Hard constraints may be defined to prevent clash between resources, for example a teacher assigned to two lessons at the same time, or to specify requirements that must be met for certain lessons, such as the necessity for double lessons or when a lesson must occur before or after other lesson (PILLAY, N., 2014).

Soft constraints define the quality of the timetable and can be contradictory. Because of that it may be impossible to satisfy all of them. A timetable that is feasible and also has a minimum number of violated soft constraints is said to be optimal (SAVINIEC et al., 2020). Soft constraints may indicate time preferences of teachers, for example to group all lessons in the morning period or to have free time after lunch (PILLAY, N., 2014). Due to differences between each country legislation, the school timetabling problem differs greatly between particular education systems (POST et al., 2012), thus the hard and soft constraints also differ drastically from one problem to the next.

2.3 Problem Solvers

The school timetabling problem is known to be NP-complete (EVEN; ITAI; SHAMIR, 1976). The difficulty of the problem arises from simple realistic conditions, such as a wide subject choice for students or the requirement for two lessons to be taught at consecutive time periods (COOPER; KINGSTON, 1996; EIKELDER; WILLEMEN, 2001). On account of its difficulty, many algorithms have been used to tackle the problem. Existing algorithms can be classified into one or more of six types (TAN et al., 2021):

- **Mathematical optimization algorithms**: Algorithms such as Integer Programming and Constraint Programming belong to this class. Integer Programming assumes the objective function and the constraints are linear, while Constraint Programming accepts some non-linearity, and they restrict some or all of the problem variables into integer values. They represent the problem as a model in which a respective solver can find the optimal solution (FONSECA, G. H. et al., 2017; TASSOPOULOS; ILIOPOULOU; BELIGIANNIS, 2020).

- **Meta-heuristic algorithms**: This class is composed of very general purpose problem-solvers designed to find an acceptable (generally feasible) solution in a reasonable amount of computational time, although they do not ensure optimality. Their design considers exploration and exploitation approaches. They can be further divided in Population-based algorithms and Single solution-based algorithms (DUTTA; SIL; DUTTA, 2020).
- **Graph coloring algorithms**: Algorithms that models the timetabling problem using graph theory to represent the problem variables (BUDIONO; WONG, 2012).
- **Matheuristics**: This approach combine the use of heuristics with the mathematical optimisation algorithms (SAVINIEC et al., 2020).
- **Hyper-heuristic**: This class uses a set of heuristics and a selection method to automate the selection of which heuristic should be applied (AHMED; ÖZCAN; KHEIRI, 2015).
- **Hybrid**: All solvers which combine the strengths of several (two or more) meta-heuristic algorithms in a unified framework are classified as Hybrid solvers (TASSOPOULOS; BELIGIANNIS, 2012; FONSECA, G. H. G. da et al., 2016).

The survey by (TAN et al., 2021) observes that Integer Programming based methods are the current state-of-the-art for three different datasets of School Timetabling Problem. It also shows that meta-heuristics are currently the most popular class of solvers used by researchers, followed by mathematical optimization methods, hyper-heuristics and then matheuristics. The majority of the meta-heuristics are non-population based methods, as one drawback of using the population-based approach is the long execution time in finding a good quality solution.

2.4 Problem Formulation

We formalize our school timetabling problem as the optimization of a cost function C defined as:

$$C = \sum_{c} C_c \tag{1}$$

where C_c is the cost of each individual constraint defined by the problem instance. We use the XML archive for High School Timetabling (XHSTT) (POST et al., 2012) as the standard data format in our work. We will describe briefly the necessary properties of the format to understand the cost function.

An instance of XHSTT is composed of four entities, namely times, resources, events and constraints. Events and times can also be part of an event group or a time group.

An event is a meeting between resources. It has associated with it a set of *event resources*, that are required for the event to happen. Event resources may be preassigned to the event, or they may be left open for a solver to assign, subject to constraints. For example, the event “Math Lesson” requires a math teacher and a class of students. The teacher and class may be defined in advance or the solver may assign them, ensuring that every class has the correct number of lessons according to the constraints of the instance.

Each constraint is of a specific type, e.g. *avoid clashes* or *assign time*. The type of the constraint defines its *points of application* and its *deviations*. A point of application is an instance entity, such as a resource or an event, in which the constraint is applied and differs for each type of constraint. Each constraint requires at least one point of application. A deviation is a non-negative integer that indicates how and to what degree a constraint is broken. The deviation is later converted into the *cost* of the constraint following a cost function of the instance.

2.4.1 Objective Function

Each constraint penalizes timetables for its events, event groups, or resources that do not meet certain characteristics. The penalty of a constraint is called a deviation and is converted into a cost. The cost of a constraint is calculated by the following formula:

$$C_c = w_c * f(d), \tag{2}$$

where $w_c \in \mathbb{N}$ is the weight of the constraint and $f(d)$ is a cost function. The cost function is applied to the deviation d, producing an integer which is multiplied by the weight to obtain the cost. The cost function f can have one of three behaviours:

- **Linear:** The deviation, unchanged
- **Quadratic:** The square of the deviation
- **Step:** 1 if the deviation is non-zero, and 0 otherwise

Each constraint has the boolean value “Required” associated with it. If Required is true, the constraint is a hard constraint and the cost is added to the infeasibility value (hard cost) of the solution. If Required is false, the constraint is a soft constraint and the cost is added to the objective value (soft cost) of the solution. The hard cost always takes priority over the soft cost, as violations of hard constraints are serious defects for a solution. Solvers aim to find solutions with very few hard constraint violations, and although the existence of such solutions is not guaranteed, realistic instances should have them.

3 QAOA ALGORITHM

This chapter is divided in two main sections. The first sections explains the necessary concepts of quantum computation to understand the QAOA algorithm. The second section explains the inner workings of the Quantum Approximate Optimization Algorithm and its variation, the Quantum Alternating Operator Ansatz.

3.1 Quantum Computation

Quantum computation is a new computing paradigm that has become more proeminent in the last few years. Its most important characteristics arise from its use of quantum mechanics properties to perform computations.

3.1.1 Qubits

Associated with every quantum system there is a complex vector space with inner product (Hilbert Space) called state space of the system. The system is completely described by its state vector, which is a unit vector from the state space. The dimension of Hilbert Space is determined by the associated physical system. The qubit is a unit vector from a bidimensional Hilbert space. We define the computational basis vectors of a qubit as the vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Any qubit can be written as a linear combination of $|0\rangle$ and $|1\rangle$. Qubits in the form $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, $\alpha, \beta \in \mathbb{C}$, are said to be in superposition when $\alpha \neq 0$ and $\beta \neq 0$. The coefficients α and β are called the amplitude of the basis state and are directly correlated with the probability of measuring the state.

In order to extract classical information from quantum computation it is necessary to measure the qubit. The measurement operation collapses the qubit to one of the vectors of the computational basis, and returns a classical bit with value 0 or 1 depending on the collapsed state. A collapsed state will always return the same outcome when measured again and any information about its previous superposition is lost.

For a given qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the probability of measuring $|0\rangle$ is $|\alpha|^2$ and $|1\rangle$ is $|\beta|^2$. As qubits are unit vectors, it follows that $\||\psi\rangle\| = 1$. Therefore, $|\alpha|^2 + |\beta|^2 = 1$. This is called the normalization condition, and it ensures that the sum of all measurement probabilities of the qubit states equals to 1.

3.1.2 Operators and Quantum Gates

The quantum circuit model describes quantum computation as the application of discrete unitary operators, called quantum gates, that act on the state space of the

qubit system. Their behaviour is described by an unitary matrix. A matrix U is unitary if $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the adjoint and I is the identity matrix. The adjoint is the transpost and complex conjugated of a matrix. A unitary matrix has the property to preserve the inner product of the vector that it operates, which guarantees that the normalization condition of the qubit is preserved after the application of the quantum gate. This is the only condition for a quantum gate, and any unitary matrix represents a valid quantum operator.

In quantum mechanics, the evolution of a quantum system in continuous time can be described by a Hermitian operator called the Hamiltonian of the system. An operator A is Hermitian, or self-adjoint, if $A = A^\dagger$. A Hermitian operator also has a matrix representation. Given the Hamiltonian H of a system, we can make a unitary operator which represents the same evolution following the exponentiation

$$U = e^{iH},$$

where i is the imaginary unit. More importantly, we can show that from a Hermitian operator, any unitary operator can be realized in this form (NIELSEN; CHUANG, 2010). This result shows a correspondence between the continuos-time description of evolution using a Hamiltonian, and the discrete-time description using quantum gates.

The act of a quantum gate on a qubit is often referred as a rotation. This name comes from the fact that any quantum state can be described as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle,$$

where $\varphi \in [0, 2\pi]$ describes the relative phase and $\theta \in [0, \pi]$ defines the probability of measuring $|0\rangle$ and $|1\rangle$. Following this description, every quantum state can be illustrated on the surface of a 3-dimensional unitary sphere called Bloch's Sphere, and quantum gates can be interpreted as rotations of the state vector aroud an axis of the sphere (NIELSEN; CHUANG, 2010).

Some of the most known operators are the Pauli matrices X, Y and Z, described in Figure 1. Each matrix acts as a rotation of 180° around an axis of the Bloch's Sphere. Some of these rotations have well-known behaviours. Most notably, the X operator swaps the amplitudes of the qubit, and when it acts on the computational basis it changes the qubit state from $|0\rangle$ to $|1\rangle$ and vice-versa. The Z operator adds a relative phase of -1 to the qubit, which changes the state $|1\rangle$ to $-|1\rangle$ while leaving $|0\rangle$ unchanged. The operators X and Z are sometimes called, respectively, the bit-flip operator and the phase-flip operator.

The relative phase of a qubit is an important concept that differenciates states within the computational basis. We say two states are said to differ by a relative phase in some basis if each of the amplitudes in that basis is related by a factor $e^{i\varphi}$, $\varphi \in \mathbb{R}$. For example, consider the states $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$. For both states the magnitude of

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Figure 1 – The Pauli matrices.

the amplitudes is the same, $\frac{1}{\sqrt{2}}$, however the amplitude of $|1\rangle$ differ in sign. In this case, each state gives rise to physically observable differences in measurement statistics, and it is not possible to regard them as physically equivalent

If both amplitudes of the state differs by the same phase we say the state has a global phase. Global phases do not change the statistics of measurement predicted for the state, for example $|\psi\rangle = 5i\,|\psi\rangle$. Thus we may ignore global phase factors as being irrelevant to the observed properties of the physical system.

3.2 QAOA

The QAOA is a quantum gate model algorithm capable of finding an approximate solution for constraint satisfaction problems (CHOI; KIM, 2019). QAOA begins with mapping the objective function to an Hamiltonian to bring the problem into Hilbert space. This problem Hamiltonian works together with a mixing Hamiltonian, which transfers probability amplitude between different basis states encoding problem solutions.

The problem and mixing Hamiltonians are then made into unitary operators parameterized by the real valued angles γ and β. Both operators are then applied alternately p times each to a suitable initial quantum state. A depth-p algorithm has $2p$ parameters (HADFIELD, 2018).

In the end, the resulting quantum state is measured and the expectation value of the problem Hamiltonian is estimated. We use an iterative process which obtains the parameters that optimize the expectation value of the problem Hamiltonian to reach the solution. The process for finding optimal parameters uses classical optimization methods. For this reason, QAOA is in the category of hybrid quantum-classical algorithms.

In its first publication by (FARHI; GOLDSTONE; GUTMANN, 2014) the QAOA was an approximate optimization algorithm, whose behaviour was based in the alternated application of a Hamiltonian based in a cost function and a Mixing Hamiltonian. Later, this framework was expanded by (HADFIELD; WANG; O'GORMAN, et al., 2019) so that it would allow families of more general operators. This extension of Quantum Approximate Optimization Algorithm into the Quantum Alternating Operator Ansatz is very significant because it enables the use of QAOA to a wide variety of approximate optimization, exact optimization, and sampling problems. The acronym was reworked intentionally so that it continues to apply to both prior and future work with the algorithm and serves the purpose to remove the confusion of using QAOA in contexts

besides approximate optimization and to remove the redundancy of the phrase “QAOA Algorithm”.

3.2.1 Quantum Approximate Optimization Algorithm

Suppose an objective function $f(z) = \sum_{k=1}^{m} f_k(z)$ to be maximized, where $z = z_1 z_2 \ldots z_n$ is a n-bit string and $f_k(z) = 1$ if z satisfies clause k and 0 otherwise. We define the problem Hamiltonian C that acts on a quantum state $|z\rangle$ as

$$C|z\rangle = \sum_{k=1}^{m} C_k |z\rangle = f(z)|z\rangle . \tag{3}$$

From Equation 3, it follows that C has eigenvectors $|z\rangle$ and eingenvalues $\sum_{k=1}^{m} C_k(z) = f(z)$. For an arbitrary state $|\psi\rangle$ we can calculate the expectation value of C, that is $\langle C\rangle = \langle\psi| C |\psi\rangle$. If $|\psi\rangle = |z\rangle$, $\langle C\rangle = \langle z| C |z\rangle = f(z)$. Therefore, finding $|z\rangle$ that can maximize $f(z)$ corresponds to finding a state $|\psi\rangle$ that maximizes the expectation value $\langle C\rangle$ (WANG; ABDULLAH, 2018).

QAOA works by rotating an initial state $|s\rangle$ to make it closer to a state $|z'\rangle$ that maximizes the objective function. The initial state is selected as an equal superposition of all possible solutions

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle , \tag{4}$$

and we define two types of unitary rotation matrices $U_C(\gamma)$ and $U_B(\beta)$.

The operator $U_C(\gamma)$ is parameterized by the angle γ and is derived from the problem Hamiltonian C from Equation 3.

$$U_C(\gamma) = e^{-i\gamma C} = \prod_{k=1}^{m} e^{-i\gamma C_k}. \tag{5}$$

This means that for a certain component $|z\rangle$, a phase $e^{-i\gamma}$ will be added in front of $|z\rangle$ for each satisfied condition C_k (WANG; ABDULLAH, 2018).

The rotation $U_C(\gamma)$ alone does not change the probability of obtaining basis states encoding different problem solutions. In order to achieve that we define the mixing Hamiltonian B as

$$B = \sum_{j=1}^{n} X_j, \tag{6}$$

where X_j is the Pauli operator X applied in the j-th qubit. The rotation operator $U_B(\beta)$, which depends on angle β is then defined as

$$U_B(\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j} \tag{7}$$

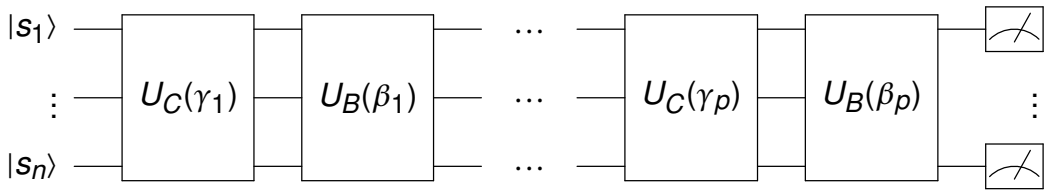

Figure 2 – QAOA circuit

It is not guaranteed that a single use of each rotation is enough to obtain an state close to $|z'\rangle$. To work through that, both $U_C(\gamma)$ and $U_B(\beta)$ are applied multiple times, with different β and γ each time. The final circuit is illustrated in Figure 2. Suppose we apply p times each operator, we obtain a new state defined as:

$$|\gamma,\beta\rangle = U_B(\beta_p)U_C(\gamma_p)\ldots U_B(\beta_1)U_C(\gamma_1)|s\rangle, \tag{8}$$

which depends on the angles γ and β and where $\gamma = (\gamma_1,\ldots,\gamma_p)$ and $\beta = (\beta_1,\ldots,\beta_p)$. It is not easy to determine γ and β in advance and there are many possible ways to find such optimal angle as this corresponds to a $2p$-dimensional optimization.

The step-by-step of QAOA worflow is as follows:

1. Begin with the initial state $|\psi\rangle = |s\rangle$.
2. (Classical computer) Initialize $2p$ parameters β and γ.
3. (Quantum computer) Construct $|\gamma,\beta\rangle$ using $U_B(\beta_p)U_C(\gamma_p)\ldots U_B(\beta_1)U_C(\gamma_1)|s\rangle$ with angles determined in the previous step.
4. (Quantum computer) Measure $|\gamma,\beta\rangle$ in computational basis set and obtain a value $|z\rangle$.
5. Use the operator $C = \sum_{k=1}^{m} C_k(z)$ to calculate $f(z)$.
6. Repeat step 1 to 4 to obtain a distribution of states $|z\rangle$. Each $|z\rangle$ corresponds to a $f(z)$ which results in a distribution of $f(z)$ with largest value $f(z')$. The expectation value of C, $\langle\gamma,\beta|\,C\,|\gamma,\beta\rangle$, is obtained by measuring the average of the distribution. We only output the $f(z')$ and the state $|z'\rangle$ as the result of applying $2p$ parameters β and γ.
7. Select a new set of $2p$ parameters β and γ and repeat step 3 to 6 as part of a $2p$-parameter function optimization where only the function evaluation involves a quantum system. In the end, obtain a distribution of $f(z')$ and choose the largest one as the final output.

3.2.2 Quantum Alternating Operator Ansatz

The algorithm developed by (HADFIELD; WANG; O'GORMAN, et al., 2019) extends the idea of (FARHI; GOLDSTONE; GUTMANN, 2014) encompassing a more general class of quantum states that can be algorithmicaly accessible. The key to this extension is about to enable the alternation using partial mixing operators which allows the algorithm to construct families of mixing operators that cannot be expressed as $e^{-i\beta B}$ for a fixed mixing Hamiltonian B. These partial mixing operators also make the implementable mixers more efficient than was possible in the original framework. For example, when the optimization is over solutions that must satisfy hard constraints, the partial mixers can restrict the search to the feasible space, instead of the full configuration space of the problem.

The Quantum Alternating Operator Ansatz can be applied to problems beyond previously possible, such as the NP-Optimization problems Traveling Salesperson Problem, Single-Machine and Multi-Processor Scheduling and Minimum Graph Coloring (HADFIELD; WANG; RIEFFEL, et al., 2017; RUAN et al., 2020).

An instance of an optimization problem is a pair (F, f), where F is the domain and $f : F \to \mathbb{R}$ is the objective function to be optimized. The domain is generaly expressed as a possible subset of a larger configuration space, specified by a set of problem conditions. In order to be implemented in current quantum hardware, each configuration space must be encoded as a subspace of a Hilbert space of a multiqubit system, with the domain corresponding to a feasible subspace of the configuration space.

Let $\mathscr{F}$ be the Hilbert space of dimension $|F|$ which the standard basis is $\{|x\rangle : x \in F\}$. A QAOA circuit is characterized by two families of operators parameterized by the real-valued angles β and γ that act in $\mathscr{F}$:

- A family of phase separation operators $U_P(\gamma)$ which depends of the objective function f;
- A family of mixing operators $U_M(\beta)$ which depends of the domain and its structure.

Specifically, a QAOA$_p$ circuit consists of p alternated applications of the operators from these two families:

$$Q_p(\beta,\gamma) = U_M(\beta_p)U_P(\gamma_p)\dots U_M(\beta_1)U_P(\gamma_1). \tag{9}$$

The quantum alternating operator ansatz consists of the states representable as the application of this circuit on the initial state $|s\rangle$:

$$|\beta,\gamma\rangle = Q_p(\beta,\gamma)\,|s\rangle\,. \tag{10}$$

3.2.2.1 QAOA Mapping

For a given optimization problem, the QAOA mapping of a problem consists of an initial state, a family of phase separation operators and a family of mixing operators. The circuit of the original quantum approximate optimization algorithm fits in this paradigm, with unitaries in the form $e^{-i\gamma C}$ as the phase separation operator and $e^{-i\beta B}$ as the mixing operator.

A suitable QAOA mapping for the problem must follow some design criteria (HADFIELD; WANG; O'GORMAN, et al., 2019) that restricts state evolution to the feasible subspace. This restriction improves the performance of the algorithm, since it does not need to search the entire configuration space for a solution. The QAOA mapping can be used to compile directly to a gate-level quantum circuit, once a problem encoding is selected. Problem encoding defines the translation of the problem parameters into qubits. Different problem encodings may lead to different gate and qubit costs. Given a domain, an encoding of a configuration space, a phase separator and a mixer, there are many compilations of the QAOA to quantum circuits.

It is required that the initial state $|s\rangle$ to be trivial to implement, which means we must be able to create it with a quantum circuit with constant depth (by the size of the problem) from the state $|0\ldots0\rangle$. The initial state can be a unique possible solution, commonly implemented by a 1-depth circuit made by single bit-flip operations. In such case the phase operator applies only a global phase to the initial state, and it is interesting to consider that the algorithm begins with a mixing operator $U_M(\beta_0)$ applied on the initial state as a first step.

These criteria can be relaxed to a logarithmic depth if necessary. However, it should not be relaxed too much, otherwise it would increase the complexity of the algorithm as a whole. Algorithms with more complex initial states should be considered hybrid algorithms, with an initialization part and a QAOA part (HADFIELD; WANG; O'GORMAN, et al., 2019).

We require that the family of phase separation operators to be diagonal in relation to the computational basis. In almost every case, we assume

$$U_P(\gamma) = e^{-i\gamma H_f}, \tag{11}$$

where H_f is the Hamiltonian of the objective function f to be optimized.

The mixing operators $U_M(\beta)$ must follow two criteria, they must preserve the feasible space and they must provide transitions between all pair of states corresponding to feasible points. That means, for every instance of the parameter β, the resulting unitary takes feasible states to feasible states and for each feasible pair of computational states $x, y \in F$ there is a parameter of value β^* and some positive integer r such that it is possible to connect states $|x\rangle$ and $|y\rangle$ by applying the mixer $U_M(\beta^*)$ r times.

4 METHOD

We propose to tackle the school timetabling problem as formalized in Chapter 2 with a Two-stage optimization algorithm using the QAOA quantum circuit presented in Chapter 3. A Two-stage optimization algorithm attempts to minimize the soft-constraints violations only after a feasible solution, i.e. a solution that follows all hard-constraints, has been reached. This method differs from an One-stage optimization algorithm, where both hard and soft constraints are considered simultaneously during the optimization process (LEWIS; PAECHTER; ROSSI-DORIA, 2007).

The QAOA algorithm is a hybrid quantum-classical algorithm that uses a parameterized quantum circuit with a classical optimization process over the quantum gates parameters. The depth of the resulting quantum circuit grows linearly with a parameter p that regulates the number of applications of the quantum circuit (FARHI; GOLDSTONE; GUTMANN, 2014), which makes QAOA a strong candidate for running in NISQ machines.

Our Two-stage optimization uses the QAOA circuit for solving the Minimum Graph Coloring problem as defined by Hadfield (HADFIELD, 2018) to address the hard constraints and later addressing the soft constraints of the timetabling problem using the classic optimization process of QAOA. It is important to notice that although our heuristic has two stages, defined by its quantum part and its classical part, we used only the QAOA algorithm. Because of that, the transition between both stages does not represent a middle point of our heuristic and during the classical optimization loop it may come back to the first stage, following the behaviour of QAOA. We detail both stages in the following sections.

4.1 First Stage - Hard Constraints

We address the hard constraints of the school timetabling problem by first reducing it to a graph coloring problem. Graph coloring is a general model for conflict resolution where each node of a graph receives a color and two adjacent nodes cannot have the same color. In the context of school timetabling the number of colors used is equal to the number of time periods necessary to schedule all classes without clashing of resources.

After extracting all the information about the resources and events of the timetabling instance, we create the conflict graph $G = (N, E)$ (see Alg. 1 line 1), where N is the set of all lessons and $E = \{(i, j) \mid i, j \in N,\ i$ cannot be scheduled at the same time as $j\}$, and $|N| = n$, $|E| = m$. Two lessons cannot be scheduled together if a constraint prevents them from doing so. For example, a constraint may require that a resource, e.g. a teacher, not to be utilized by more than one lesson at the same time. The conflict graph summarizes all the hard-constraints of the problem.

Algorithm 1: QAOA for School Timetabling Problem

```
   Input  : XHSTT archive tt−instance, Number of time periods k, QAOA parameter p
1  G ← createConflictGraph(tt−instance)
   // QAOA algorithm preparation
2  initialState ← colorGraph(G, k)
3  β[1..p], γ[1..p] ← INTERP()
   // Classic Optimization Process over the β and γ parameter vectors
4  while don't stop do
5      finalResult ← minimize(QAOACircuitEvaluation(β, γ, initialState))
6  return finalResult
```

When the graph is ready, we start to prepare the QAOA algorithm. We first prepare the initial state for QAOA (line 2). We color the conflict graph with at most k colors, where k is the number of available time periods for the lessons to be assigned. The available time periods were extracted in advance accordingly with the instance description. The initial state must be a valid coloring, otherwise the QAOA will search for solutions outside feasible space. For the QAOA to guarantee that the final solution is feasible the coloring of the initial state must preserve all hard-constraints.

We also initialize the parameter vectors γ and β (line 3). Random initialization, multistart optimization (SHAYDULIN; SAFRO; LARSON, 2019), or the use of heuristics are a few examples of strategies used to initialize them. For our parameter setting strategy, we adapted the INTERP heuristic strategy described in (ZHOU et al., 2020) which uses linear interpolation to produce a good initial point for optimizing the QAOA as one iteratively increases its depth.

After defining the initial state and the parameter vectors γ and β, the QAOA circuit is created (line 5, calling Algorithm 2). The total cost of the quantum algorithm is $O(p(k^2m+nk))$ basic quantum gates, where p is the QAOA parameter. Our implementation uses $nk+n$ qubits. This value differs from the original circuit (HADFIELD, 2018) because we increased the number of ancillas qubits in order to reduce the circuit depth.

Inside Algorithm 2, after the measurement (line 4), the final state will collapse to a vector $|z\rangle$ where z is a valid graph coloring. We then begin the second phase by passing $|z\rangle$ to the objective function we want to optimize (Alg 2, line 5).

4.2 Second Stage - Soft Constraints

Our solution addresses the soft constraints using the classical optimization loop of the QAOA algorithm, shown in Algorithm 1 (line 4-5). Usually, the phase separator U_P encodes the function to be optimized, which in the case of Minimum Graph Coloring, is the function that counts how many colors were used in the coloring. However, we changed the objective function so that it represents our soft constraints while keeping the same phase separator. This alteration changes significantly the inner working of

Algorithm 2: QAOA Quantum Circuit Evaluation

```
   Input  :parameter vector β, parameter vector γ, graph coloring initialState
1  results ← []
2  QAOACircuit ← constructQuantumCircuit(β, γ, initialState)
3  for j ← 1 to 10000 do
4  |   measurement ← (measure(QAOACircuit))
   |   // ObjectiveFunction can be seen in Equation (12)
5  |   results.append(ObjectiveFunction(measurement))
6  expectedValue ← results.average()
7  return expectedValue
```

QAOA, because the quantum circuit normally looks for an answer that already optimizes the objective function.

Even though the objective function is not encoded by the phase separation operator, the solutions found by the quantum circuit are still valid solutions, since the timetabling problem can be reduced to the graph coloring problem. Therefore, while the QAOA circuit searches for a solution that uses the least amount of colors, keeping the solution within feasible space, the classical optimization loop guides the parameters so that the solution fails the least amount of soft constraints. To compensate for this change, we expect that longer optimization runs will be needed.

Our new objective function that counts the number of soft constraints violations receives as input a valid coloring and is described as the weighted sum:

$$\sum_{c \in C} w_c f(c) \tag{12}$$

where C is the set of all soft-constraints and w_c is the weight associated with the constraint described in the problem instance. If the constraint was violated then f(c) is 1, and 0 otherwise.

The function evaluation for our $2p$-parameter function optimization is described inside Algorithm 2. After the measurement of the QAOA quantum circuit (line 4), the result of the measurement is passed to the objective function (line 5). This process must be repeated in order to create a state distribution and the expected value to be estimated (line 6). The expected value is then returned to the classical optimization process (line 7). The expected value represents the average result returned by QAOA.

4.3 Classical Optimization Routine

The optimization loop of QAOA follows the behaviour of the chosen classical optimization algorithm, which dictates aspects such as convergence guarantee and the stop criteria.

We use the algorithms COBYLA (POWELL, 1994) and CMA-ES (HANSEN, 2007) as the optimization routines to work with QAOA. Hybrid quantum-classical al-

gorithms work well with a variety of classical optimization algorithms. The most commom approachs are black-box gradient-free (GUERRESCHI; MATSUURA, 2019; SHAYDULIN; ALEXEEV, 2019; VIKSTÅL et al., 2020; UTKARSH; BEHERA; PANIGRAHI, 2020), or gradient-based (ZHOU et al., 2020) optimizations. Other strategies, such as evolutionary strategies (ROCH et al., 2020), reinforcement learning (KHAIRY et al., 2020), and machine learning (ALAM; ASH-SAKI; GHOSH, 2020) have also been used with QAOA. A recent work (FERNÁNDEZ-PENDÁS et al., 2021) attempts an exhaustive comparison between different optimization strategies with QAOA and concluded that COBYLA is capable of finding good quality solutions while still being faster than other tested approaches. We chose CMA-ES as an alternative to COBYLA to ascertain the behaviour of QAOA with other optimization routines.

5 RESULTS

We simulated our quantum circuits with the Parallel-Bitwise-Simulator for the Ket Quantum Programming Language (DA ROSA; DE SANTIAGO, 2021). We used an Intel Xeon E5-2640 with 10 nodes and hyperthreading, 128Gb of RAM and an NVIDIA Tesla K40c GPU.

We have performed experiments by using the XHSTT Dataset Denmark-SmallSchool instance described in Table 1. The instance asks to schedule a total of 25 lessons in 4 time periods. The school has a total of 74 students and 9 teachers. Each lesson is composed of a subset of students and a teacher defined in advance. There are three constraints in this instance:

- AssignTimeConstraint: hard constraint, demands that all lessons must be assigned to a time period.
- AvoidClashesConstraint: hard constraint, ensures that there is no clashes between resources.
- PreferTimesConstraint: soft constraint, indicates preferable time periods for specific lessons. There are only 4 lessons which have a preferable time period to be scheduled.

All constraints have a weight of 1 associated with it.

Table 1 – XHSTT Dataset Denmark-SmallSchool

Assets	Times	Teachers	Rooms	Students	Classes	# events	Total duration
Values	4	9	-	74	-	25	250

The resulting conflict graph with each lesson is shown in Figure 3. We used a coloring heuristic to find a suitable 4-coloration for the graph to use as initial state. Our initial state scored a 4 in the objective function, which means it broke all soft-constraints. This instance required a total amount of 125 qubits to simulate.

We evaluated our method for paramater p assuming values 1,2,4 and 8. We made a total of 20 runs for each value using COBYLA as the optimization algorithm and only a single run using CMA-ES. Each run used the same initial state. As a parameter setting strategy we adapted the INTERP heuristic described in (ZHOU et al., 2020), so instead of generating the next $p+1$ parameters after a local optima was found, we used linear interpolation to generate good initial values for the next $2p$ parameters

Because of the restricted size of RAM in our GPU card, we limited the states representable by our simulator to only states with a probability greater than 10^{-6}, because these states are more likely to be chosen by a limited number of measurements.

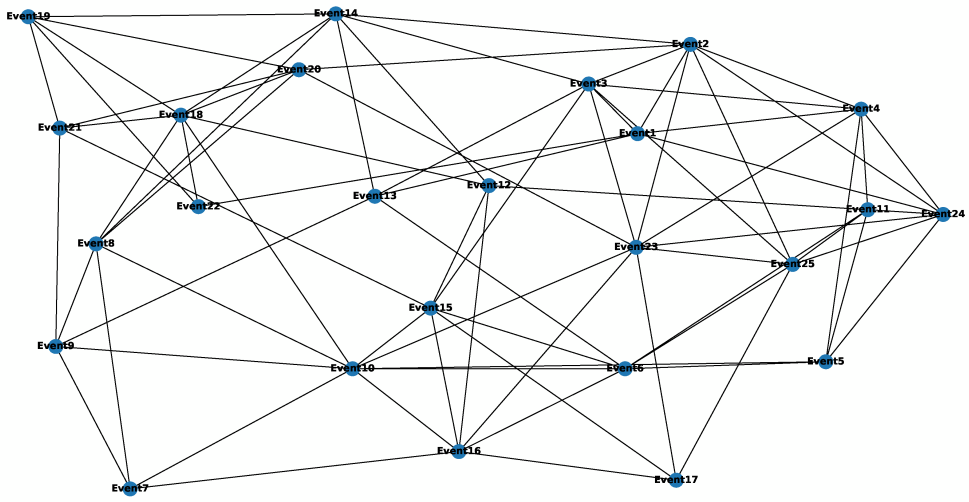

Figure 3 – Conflict graph of instance Denmark-Smallschool.

We believe the ansatz obtained with higher values of p will be the most affected by this restriction, since they reach the most states.

We found that our method was unable to optimize the SmallSchool instance as originally stated. Our initial state had 4 colors and 4 is the chromatic number of the conflict graph. As our QAOA mixer works by swapping the colors of two nodes if this swap does not result in any conflict with the neighbouring nodes, it was unable to search new solutions since any swap would result in a conflict. As a consequence of these results we developed two new instances that we called Den-aux and Den-5. Den-aux adds two auxiliary nodes to our conflict graph to allow the mixer to search for new solutions and Den-5 is a relaxation of the original instance, in which we add a fifth color that represents an extra time period in the timetable.

The following experiments were made using only COBYLA as the optimization algorithm. The number of function calls of CMA-ES greatly exceed that of COBYLA and as consequence the time needed for a single run of CMA-ES was greater than all the runs of COBYLA together. As pointed out in (FERNÁNDEZ-PENDÁS et al., 2021) due to the probabilistic nature of QAOA, the algorithm may have more chances of obtaining better values if we use a fast optimization algorithm and run our heuristic several times.

5.1 Instance Den-aux

We added two additional nodes to the graph of the instance Denmark-Smallschool to attempt to solve the original problem with 4 time periods. These auxiliary nodes were connected with all original nodes and were colored with two new colors. This change allowed our QAOA circuit to mix the coloring and to find new solutions since every node was able to exchange colors with any of the new auxiliary nodes. At the end of our

method we processed the conflict graph as to remove the auxiliary nodes and convert the graph in a suitable schedule for the timetabling problem with only 4 time periods and 25 lessons. The resulting conflict graph with each lesson is shown in Figure 4. This instance required a total amount of 189 qubits to simulate.

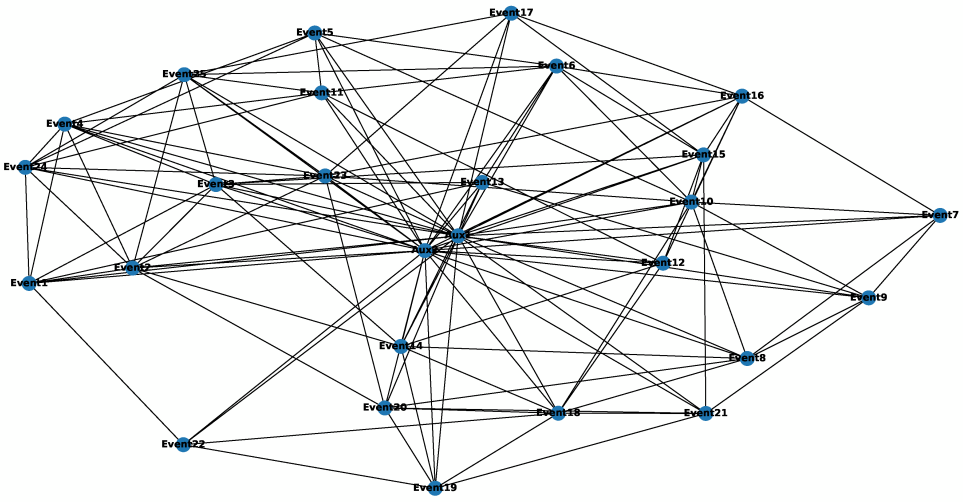

Figure 4 – Conflict graph of instance Denmark-Smallschool with auxiliary nodes.

We ran our heuristic 10 times for all values of p. We limited the maximum number of function evaluations of COBYLA to 100 for parameter p 1,2 and 4 and 50 for $p = 8$. We also further limited the states of $p = 8$ for only those with probability greater than 10^{-4}, due to our GPU not being able to handle greater precision.

We noticed during the post-processing of the conflict graph that some solutions achieved by the method were invalid. In some occasions both auxiliary nodes took on the same color and as consequence the main nodes of the graph had a final coloring with 5 colors, which should be impossible since the instance has only 4 time periods. These faulty solutions were discarded by the method, that kept only the solutions with 4 colors.

Even with the auxiliary nodes, our heuristic was unable to find better results with high probability for the Denmark-Smallschool instance as originally stated. On average for $p = 1$ we obtained a cost value of 3.976, which is a small improvement from the 4 of the initial state. As the value of p increased we improved our cost value in 0.01. However, we noticed that for $p = 8$ the results obtained were worse than for $p = 4$. This is mostly due to the fact that the length of the runs for $p = 8$ were shorter than the rest and the number of function evaluations was not enough for the optimization algorithm to converge. All the results are plotted in Figure 5.

We chose the best run for each value of parameter p and plotted the function distributions in Figure 6. The histograms show that less than 6% of all results for all

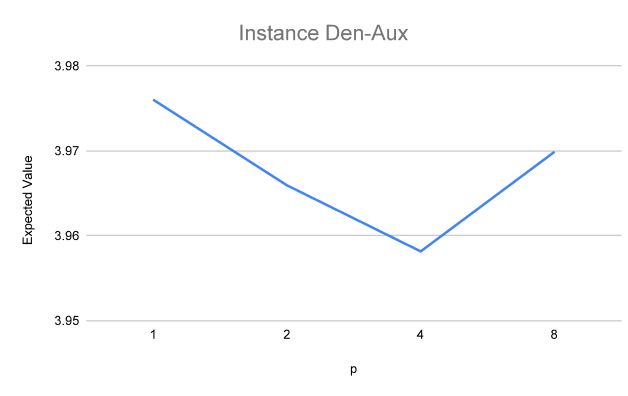

Figure 5 – Average expected value from instance Den-aux.

values of p obtained by our heuristic for the instance Den-aux showed an improvement from the 4 scored by our initial state.

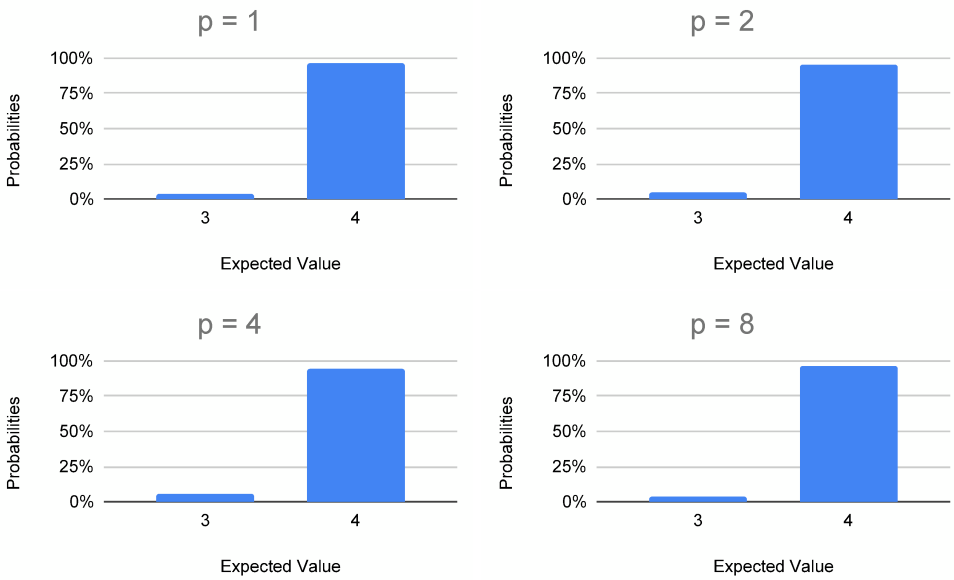

Figure 6 – Objective function distribution from instance Den-aux.

5.2 Instance Den-5

We developed a new instance by adding a fifth time period to our timetable. We did not change the constraints or the number of lessons, so our new instance uses the same conflict graph described in Figure 3 and the same initial state obtained previously. The additional color allowed our QAOA circuit to search for new solutions, because every node could swap its own color with the fifth new color without having any conflict with its neighbours. This new instance required a total amount of 150 qubits to simulate.

For this instance we also ran our heuristic 10 times for each value of p and used the same limitations to function evaluations of COBYLA and representable states for

$p = 8$.

Our heuristic obtained significantly better results when optimizing the relaxed instance. For $p = 1$ the average value of the objective function was 2.971. The average results improved as the value of parameter p increased reaching the best results for $p = 4$ with a function value of 2.167. We observed again that $p = 8$ performed worse due to the limited number of function evaluations of COBYLA. The plotted results can be seen in Figure 7.

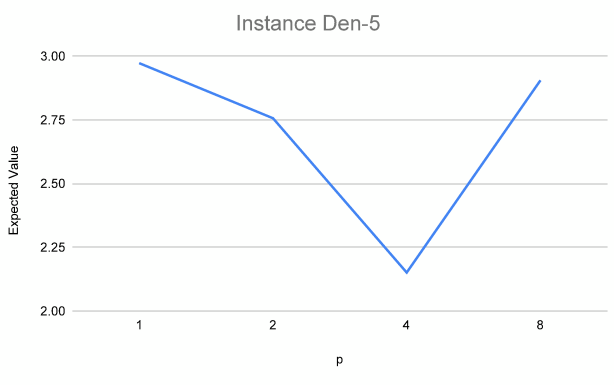

Figure 7 – Average expected value from instance Den-5.

We also analysed the function distributions for the best runs of the Den-5 instance. The histograms of Figure 8 show that our heuristic converged for a function value of 2 as the parameter p of QAOA increased. The optimal result is known to have function value of 0 so it indicates that our heuristic converged to a local optima.

Despite not finding the best result, our heuristic was able to converge to an optimized result with parameter p as low as 2. This could indicate its potential for achieving good results even within a low-depth regime.

5.3 Chapter Considerations

From the three instances analysed, our heuristic showed difficulties in optimizing the two instances which initial state had used all available time periods. This behaviour may indicate that it will also have difficulties in optimizing hard instances, i.e. timetabling problems in which all periods must be utilised with very little or no options for each allocation. However, when considering the relaxed instance, it was able to converge to an optimized result even with low values of p.

Our choice of initial state may have influenced the algorithm to fall into a local optima. The issue of trainability is known for Variational Quantum Algorithms (CEREZO et al., 2020), and there has been great research efforts focused on avoiding or mitigating the effect of Barren Plateau, which is the phenomenon where, as the number of qubits increases, the landscape of the cost function becomes flatter. One approach to the

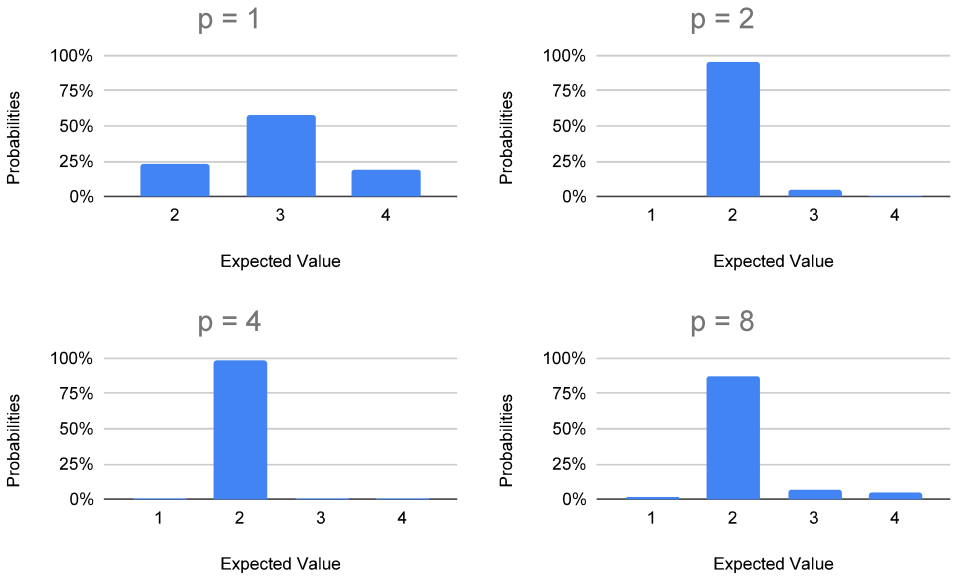

Figure 8 – Objective function distribution from instance Den-5.

problem is choosing better parameter initialization strategies (GRANT et al., 2019; ZHOU et al., 2020), as randomly initializing an ansatz can lead to the algorithm starting far from the solution, near a local minima, or even in a region with barren plateaus. We believe that together with a parameter initialization strategy, the choice of better initial states should also be considered.

6 CONCLUSION

This work proposes the use of the Quantum Approximate Optimization Algorithm as a heuristic to solve the School Timetabling Problem. We developed a Two-stage optimization process, where on the first stage the QAOA quantum circuit for the Minimum Graph Coloring problem addresses the hard constraints of the timetable and on the second stage the classical optimization loop of QAOA addresses the soft constraints. We consider this research and its findings a seminal work in using QAOA as a heuristic for the timetabling problem.

We tested our method using three different instances. The instance Denmark-Smallschool from XHSTT Dataset and two derived instances, Den-Aux and Den-5. We simulated our circuits in a noiseless environment using the Parallel Bitwise Simulator for the Ket Quantum Programming Language(DA ROSA; DE SANTIAGO, 2021). Our instances required up to 189 qubits to simulate.

Our heuristic was unable to optimize the original Denmark-Smallschool instance due to the QAOA mixer being unable to reach different solutions. The initial state of the instance started with a 4-coloring and coincidentally this is the chromatic number of the conflict graph. As consequence, the QAOA circuit could not search new solutions as it was starting from an optimal coloring. We believe this behaviour may indicate that our heuristic will also have difficulties in optimizing hard instances of the Timetabling Problem.

We then created two derived instances to further analyze our heuristic. The instance Den-Aux attempted to solve the original timetabling problem by adding two auxiliary nodes to the conflict graph and the instance Den-5 relaxed the original problem to allow a fifth time period, and as consequence a fifth color, to the conflict graph.

Our heuristic was not able to find better results than the initial state for the instance Den-Aux, as our attempt to use auxiliary nodes was not able to overcome the limitations of our heuristic. However, for the relaxed instance of Den-5 our heuristic was able to converge to an optimized result with parameter $p = 2$. This could indicate the potential of using this heuristic in near-term devices as the depth needed by QAOA was low.

6.1 Future Works

For future works our heuristic may be improved by testing the following changes:

- The heuristic can be changed from a Two-Stage optimization to an One-Stage optimization if the objective function of the soft constraints can be directly made into a phase Hamiltonian. This can help to reduce the number of runs necessary for the optimization algorithm to converge.

- The impact that the initial state has on the search for a solution can be analyzed and a strategy for choosing a suitable initial state could be devised.
- As an alternative to the graph-coloring method, the QAOA could be used to solve the timetabling problem modeled as a constraint-satisfaction problem. This alternative could be easier to simulate because a n-bit string that composes the configuration space of this problem could be mapped to n qubits, while for the graph coloring the number of qubits used scale faster, since it depends on the product of the number of events and the number of time periods.

REFERENCES

AHMED, Leena N.; ÖZCAN, Ender; KHEIRI, Ahmed. Solving high school timetabling problems worldwide using selection hyper-heuristics. **Expert Systems with Applications**, v. 42, n. 13, p. 5463–5471, 2015. ISSN 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.02.059. Available from: https://www.sciencedirect.com/science/article/pii/S0957417415001670.

ALAM, Mahabubul; ASH-SAKI, Abdullah; GHOSH, Swaroop. Accelerating Quantum Approximate Optimization Algorithm using Machine Learning. In: 2020 Design, Automation Test in Europe Conference Exhibition (DATE). [S.l.: s.n.], 2020. P. 686–689. DOI: 10.23919/DATE48585.2020.9116348.

ARUTE, Frank et al. Quantum supremacy using a programmable superconducting processor. **Nature**, v. 574, p. 505–510, 7779 Oct. 2019. ISSN 1476-4687. DOI: 10.1038/s41586-019-1666-5. Available from: https://doi.org/10.1038/s41586-019-1666-5.

AZIZ, Nurul Liyana Abdul; AIZAM, Nur Aidya Hanum. A brief review on the features of university course timetabling problem. **AIP Conference Proceedings**, v. 2016, n. 1, p. 020001, 2018. DOI: 10.1063/1.5055403. eprint: https://aip.scitation.org/doi/pdf/10.1063/1.5055403. Available from: https://aip.scitation.org/doi/abs/10.1063/1.5055403.

BELIGIANNIS, Grigorios N. et al. Applying evolutionary computation to the school timetabling problem: The Greek case. **Computers & Operations Research**, v. 35, n. 4, p. 1265–1280, 2008. ISSN 0305-0548. DOI: https://doi.org/10.1016/j.cor.2006.08.010. Available from: https://www.sciencedirect.com/science/article/pii/S0305054806001973.

BUDIONO, Tri A.; WONG, Kok Wai. A pure graph coloring constructive heuristic in timetabling. In: 2012 International Conference on Computer Information Science (ICCIS). [S.l.: s.n.], 2012. P. 307–312. DOI: 10.1109/ICCISci.2012.6297259.

CEREZO, M. et al. **Variational Quantum Algorithms**. [S.l.: s.n.], 2020. arXiv: 2012.09265 [quant-ph].

CHOI, Jaeho; KIM, Joongheon. A Tutorial on Quantum Approximate Optimization Algorithm (QAOA): Fundamentals and Applications. In: 2019 International Conference on Information and Communication Technology Convergence (ICTC). [S.l.: s.n.], 2019. P. 138–142. DOI: 10.1109/ICTC46691.2019.8939749.

COLBOURN, Charles J.; OORSCHOT, Paul C. van. Applications of Combinatorial Designs in Computer Science. **ACM Comput. Surv.**, Association for Computing

Machinery, New York, NY, USA, v. 21, n. 2, p. 223–250, June 1989. ISSN 0360-0300. DOI: 10.1145/66443.66446. Available from: https://doi.org/10.1145/66443.66446.

COOPER, Tim B.; KINGSTON, Jeffrey H. The complexity of timetable construction problems. In: **Practice and Theory of Automated Timetabling**. Ed. by Edmund K. Burke and Peter Ross. [S.l.]: Springer Berlin Heidelberg, 1996. v. 1153, p. 281–295. (Lecture Notes in Computer Science). ISBN 978-3-540-61794-5. DOI: 10.1007/3-540-61794-9_66. Available from: http://dx.doi.org/10.1007/3-540-61794-9%5C_66.

DA ROSA, Evandro Chagas Ribeiro; DE SANTIAGO, Rafael. Ket Quantum Programming. **J. Emerg. Technol. Comput. Syst.**, Association for Computing Machinery, New York, NY, USA, v. 18, n. 1, Oct. 2021. ISSN 1550-4832. DOI: 10.1145/3474224. Available from: https://doi.org/10.1145/3474224.

DE WERRA, D. An introduction to timetabling. **European Journal of Operational Research**, v. 19, n. 2, p. 151–162, 1985. ISSN 0377-2217. DOI: https://doi.org/10.1016/0377-2217(85)90167-5. Available from: https://www.sciencedirect.com/science/article/pii/0377221785901675.

DUTTA, Dipankar; SIL, Jaya; DUTTA, Paramartha. A bi-phased multi-objective genetic algorithm based classifier. **Expert Systems with Applications**, v. 146, p. 113163, 2020. ISSN 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.113163. Available from: https://www.sciencedirect.com/science/article/pii/S0957417419308814.

EIKELDER, Huub; WILLEMEN, R. Some Complexity Aspects of Secondary School Timetabling Problems. In: PRACTICE and Theory of Automated Timetabling III. [S.l.]: Springer Berlin Heidelberg, Aug. 2001. P. 18–27. DOI: 10.1007/3-540-44629-X_2.

EVEN, S.; ITAI, A.; SHAMIR, A. On the Complexity of Timetable and Multicommodity Flow Problems. **SIAM Journal on Computing**, v. 5, n. 4, p. 691–703, 1976. ISSN 0097-5397. DOI: 10.1137/0205048.

FARHI, Edward; GOLDSTONE, Jeffrey; GUTMANN, Sam. **A Quantum Approximate Optimization Algorithm**. [S.l.: s.n.], 2014. arXiv: 1411.4028 [quant-ph].

FERNÁNDEZ-PENDÁS, Mario et al. A study of the performance of classical minimizers in the Quantum Approximate Optimization Algorithm. **Journal of Computational and Applied Mathematics**, p. 113388, 2021. ISSN 0377-0427. DOI: https://doi.org/10.1016/j.cam.2021.113388. Available from: https://www.sciencedirect.com/science/article/pii/S0377042721000078.

FONSECA, George H.G.; SANTOS, Haroldo G.; CARRANO, Eduardo G. Integrating matheuristics and metaheuristics for timetabling. **Computers & Operations Research**,

v. 74, p. 108–117, 2016. ISSN 0305-0548. DOI: 10.1016/j.cor.2016.04.016. Available from: http://dx.doi.org/10.1016/j.cor.2016.04.016.

FONSECA, George H.G. et al. Integer programming techniques for educational timetabling. **European Journal of Operational Research**, v. 262, n. 1, p. 28–39, 2017. ISSN 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.03.020. Available from: https://www.sciencedirect.com/science/article/pii/S0377221717302242.

FONSECA, George Henrique Godim da et al. GOAL solver: a hybrid local search based solver for high school timetabling. **Annals of Operations Research**, v. 239, p. 77–97, 1 2016. ISSN 1572-9338. DOI: 10.1007/s10479-014-1685-4. Available from: https://doi.org/10.1007/s10479-014-1685-4.

GIDNEY, Craig; EKERÅ, Martin. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. **Quantum**, Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, v. 5, p. 433, Apr. 2021. ISSN 2521-327X. DOI: 10.22331/q-2021-04-15-433. Available from: http://dx.doi.org/10.22331/q-2021-04-15-433.

GRANT, Edward et al. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. **Quantum**, Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, v. 3, p. 214, Dec. 2019. ISSN 2521-327X. DOI: 10.22331/q-2019-12-09-214. Available from: http://dx.doi.org/10.22331/q-2019-12-09-214.

GUERRESCHI, G. G.; MATSUURA, A. Y. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. **Scientific Reports**, v. 9, p. 6903, 1 May 2019. ISSN 2045-2322. DOI: 10.1038/s41598-019-43176-9. Available from: https://doi.org/10.1038/s41598-019-43176-9.

HADFIELD, Stuart. **Quantum Algorithms for Scientific Computing and Approximate Optimization**. 2018. PhD thesis. arXiv: 1805.03265 [quant-ph].

HADFIELD, Stuart; WANG, Zhihui; O'GORMAN, Bryan, et al. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. **Algorithms**, MDPI AG, v. 12, n. 2, p. 34, Feb. 2019. ISSN 1999-4893. DOI: 10.3390/a12020034. Available from: http://dx.doi.org/10.3390/a12020034.

HADFIELD, Stuart; WANG, Zhihui; RIEFFEL, Eleanor G., et al. Quantum Approximate Optimization with Hard and Soft Constraints. In: PROCEEDINGS of the Second International Workshop on Post Moores Era Supercomputing. Denver, CO, USA: Association for Computing Machinery, 2017. (PMES'17), p. 15–21. DOI: 10.1145/3149526.3149530. Available from: https://doi.org/10.1145/3149526.3149530.

HANSEN, Nikolaus. The CMA Evolution Strategy: A Comparing Review. In: [s.l.: s.n.], June 2007. v. 192, p. 75–102. ISBN 978-3-540-29006-3. DOI: 10.1007/3-540-32494-1_4.

HILTON, A.J.W. School Timetables. In: HANSEN, P. (Ed.). **Annals of Discrete Mathematics (11)**. [S.l.]: North-Holland, 1981. v. 59. (North-Holland Mathematics Studies). P. 177–188. DOI: https://doi.org/10.1016/S0304-0208(08)73465-0. Available from: https://www.sciencedirect.com/science/article/pii/S0304020808734650.

KHAIRY, Sami et al. Learning to Optimize Variational Quantum Circuits to Solve Combinatorial Problems. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 34, n. 03, p. 2367–2375, Apr. 2020. DOI: 10.1609/aaai.v34i03.5616. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/5616.

LEWIS, Rhydian; PAECHTER, Ben; ROSSI-DORIA, Olivia. Metaheuristics for University Course Timetabling. In: **Evolutionary Scheduling**. Ed. by Keshav P. Dahal, Kay Chen Tan and Peter I. Cowling. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. P. 237–272. ISBN 978-3-540-48584-1. DOI: 10.1007/978-3-540-48584-1_9. Available from: https://doi.org/10.1007/978-3-540-48584-1_9.

NIELSEN, Michael A.; CHUANG, Isaac L. **Quantum Computation and Quantum Information: 10th Anniversary Edition**. Cambridge: Cambridge University Press, 2010. ISBN 9781107002173. DOI: 10.1017/CBO9780511976667.

PERUZZO, Alberto et al. A variational eigenvalue solver on a photonic quantum processor. **Nature Communications**, Springer Science and Business Media LLC, v. 5, n. 1, July 2014. ISSN 2041-1723. DOI: 10.1038/ncomms5213. Available from: http://dx.doi.org/10.1038/ncomms5213.

PILLAY, N. A survey of school timetabling research. **Annals of Operations Research**, v. 218, p. 261–193, July 2014. DOI: 10.1007/s10479-013-1321-8.

PILLAY, Nelishia. A Comparative Study of Hyper-Heuristics for Solving the School Timetabling Problem. In: PROCEEDINGS of the South African Institute for Computer Scientists and Information Technologists Conference. East London, South Africa: Association for Computing Machinery, 2013. (SAICSIT '13), p. 278–285. DOI: 10.1145/2513456.2513466. Available from: https://doi.org/10.1145/2513456.2513466.

PIRES, O. M. **QAOA for School Timetabling**. [S.l.]: GitLab, 2021. https://gitlab.com/OttoMP/school-timetable.

PIRES, Otto Menegasso; SANTIAGO, Rafael de; MARCHI, Jerusa. Two Stage Quantum Optimization for the School Timetabling Problem. In: 2021 IEEE Congress

on Evolutionary Computation (CEC). [S.l.: s.n.], 2021. P. 2347–2353. DOI: 10.1109/CEC45853.2021.9504701.

POST, Gerhard et al. An XML format for benchmarks in High School Timetabling. **Annals of Operations Research**, v. 194, n. 1, p. 385–397, Apr. 2012. ISSN 1572-9338. DOI: 10.1007/s10479-010-0699-9. Available from: https://doi.org/10.1007/s10479-010-0699-9.

POWELL, M. J. D. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In: **Advances in Optimization and Numerical Analysis**. Ed. by Susana Gomez and Jean-Pierre Hennart. Dordrecht: Springer Netherlands, 1994. P. 51–67. ISBN 978-94-015-8330-5. DOI: 10.1007/978-94-015-8330-5_4. Available from: https://doi.org/10.1007/978-94-015-8330-5_4.

PRESKILL, John. Quantum Computing in the NISQ era and beyond. **Quantum**, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 2, p. 79, Aug. 2018. ISSN 2521-327X. DOI: 10.22331/q-2018-08-06-79. Available from: https://doi.org/10.22331/q-2018-08-06-79.

RAGHAVJEE, Rushil; PILLAY, Nelishia. Evolving solutions to the school timetabling problem. In: 2009 World Congress on Nature Biologically Inspired Computing (NaBIC). [S.l.: s.n.], 2009. P. 1524–1527. DOI: 10.1109/NABIC.2009.5393667.

RIBIĆ, Samir; TURČINHOŽIĆ, Razija; MURATOVIĆ-RIBIĆ, Amela. Modelling constraints in school timetabling using integer linear programming. In: 2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT). [S.l.: s.n.], 2015. P. 1–6. DOI: 10.1109/ICAT.2015.7340525.

ROCH, Christoph et al. Cross Entropy Hyperparameter Optimization for Constrained Problem Hamiltonians Applied to QAOA. In: 2020 International Conference on Rebooting Computing (ICRC). [S.l.: s.n.], 2020. P. 50–57. DOI: 10.1109/ICRC2020.2020.00009.

RUAN, Yue et al. **Quantum approximate algorithm for NP optimization problems with constraints**. [S.l.: s.n.], 2020. arXiv: 2002.00943 [quant-ph].

SAVINIEC, Landir et al. Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems. **European Journal of Operational Research**, v. 280, n. 3, p. 1064–1081, 2020. ISSN 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2019.08.001. Available from: https://www.sciencedirect.com/science/article/pii/S0377221719306538.

SCHAERF, A. A Survey of Automated Timetabling. **Artificial Intelligence Review**, v. 13, p. 87–127, 1999. DOI: 10.1023/A:1006576209967.

SHAYDULIN, Ruslan; ALEXEEV, Yuri. Evaluating Quantum Approximate Optimization Algorithm: A Case Study. In: 2019 Tenth International Green and Sustainable Computing Conference (IGSC). [S.l.: s.n.], 2019. P. 1–6. DOI: 10.1109/IGSC48788.2019.8957201.

SHAYDULIN, Ruslan; SAFRO, Ilya; LARSON, Jeffrey. Multistart Methods for Quantum Approximate optimization. **2019 IEEE High Performance Extreme Computing Conference (HPEC)**, IEEE, Sept. 2019. DOI: 10.1109/hpec.2019.8916288. Available from: http://dx.doi.org/10.1109/HPEC.2019.8916288.

TAN, Joo Siang et al. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. **Expert Systems with Applications**, v. 165, p. 113943, 2021. ISSN 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113943. Available from: https://www.sciencedirect.com/science/article/pii/S0957417420307314.

TASSOPOULOS, Ioannis X.; BELIGIANNIS, Grigorios N. A hybrid particle swarm optimization based algorithm for high school timetabling problems. **Applied Soft Computing**, v. 12, n. 11, p. 3472–3489, 2012. ISSN 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2012.05.029. Available from: https://www.sciencedirect.com/science/article/pii/S1568494612002876.

TASSOPOULOS, Ioannis X.; ILIOPOULOU, Christina A.; BELIGIANNIS, Grigorios N. Solving the Greek school timetabling problem by a mixed integer programming model. **Journal of the Operational Research Society**, Taylor & Francis, v. 71, n. 1, p. 117–132, 2020. DOI: 10.1080/01605682.2018.1557022. eprint: https://doi.org/10.1080/01605682.2018.1557022. Available from: https://doi.org/10.1080/01605682.2018.1557022.

TURCINHODZIC, Razija et al. School Timetable - suggestions for a more efficient representation and a solution for schools with shifts. In: 2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT). [S.l.: s.n.], 2019. P. 1–6. DOI: 10.1109/ICAT47117.2019.8939007.

UTKARSH; BEHERA, Bikash K.; PANIGRAHI, Prasanta K. **Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm**. [S.l.: s.n.], 2020. arXiv: 2002.01351 [quant-ph].

VEENSTRA, Marjolein; VIS, Iris F.A. School timetabling problem under disturbances. **Computers & Industrial Engineering**, v. 95, p. 175–186, 2016. ISSN 0360-8352. DOI: https://doi.org/10.1016/j.cie.2016.02.011. Available from: https://www.sciencedirect.com/science/article/pii/S0360835216300389.

VIKSTÅL, Pontus et al. Applying the Quantum Approximate Optimization Algorithm to the Tail-Assignment Problem. **Physical Review Applied**, American Physical Society (APS), v. 14, n. 3, Sept. 2020. ISSN 2331-7019. DOI: 10.1103/physrevapplied.14.034009. Available from: http://dx.doi.org/10.1103/PhysRevApplied.14.034009.

WANG, Qingfeng; ABDULLAH, T. An Introduction to Quantum Optimization Approximation Algorithm. In.

WERRA, Dominique de; KOBLER, Daniel. Graph Coloring Problems. In: PARADIGMS of Combinatorial Optimization. [S.l.]: John Wiley & Sons, Ltd, 2014. chap. 10, p. 265–310. ISBN 9781119005353. DOI: https://doi.org/10.1002/9781119005353.ch10. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119005353.ch10. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119005353.ch10.

ZHOU, Leo et al. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. **Physical Review X**, American Physical Society (APS), v. 10, n. 2, June 2020. ISSN 2160-3308. DOI: 10.1103/physrevx.10.021067. Available from: http://dx.doi.org/10.1103/PhysRevX.10.021067.